

CS305 作業系統概論 Prog. #1 Proc. Generation & IPC

2022.03.07

一、作業目的

熟悉如何利用fork()系統呼叫產生新的process，以及process彼此之間如何使用POSIX shared memory互相傳遞資料。

二、作業內容

【Battleship對戰的模擬】由於世界局勢不穩定，N國和J國最近在P海峽偶爾出現雙方船艦對峙的情況。面對劍拔弩張的態勢，L上尉前來赫赫有名的海豚書院尋求成大師的協助，使用VR來演習訓練雙方船艦對戰的情況。成大師決定帶領他的高徒們先開發一個對戰的雛型系統，成為後續系統的核心。

基本要求：

這個模擬船艦對戰的過程是由一個父行程產生一個子行程（只有一個子行程），然後雙方分別在自己一個4×4的地區中任意擺放一艘2×1的砲艇，輪流炸射對方。模擬過程中，4×4的地區左上角座標為(0,0)，右下角座標為(3,3)。程式需要能從命令列讀入三個參數，第一個參數p1與第二個參數p2用來分別指定父行程與子行程的亂數種子(0~99的整數)，以及第三個參數p3表示基本功能(0)，或是進階功能(1)。

執行的流程如下：

1. 程式執行後，父行程產生一個子行程。
2. 父行程與子行程分別在自己的4×4的地區中佈署一艘2×1的砲艇。砲艇位置以隨機方式決定，決定後雙方印出位置，但此位置不可告訴對方，也就是對方無法看到自己的地圖資訊。
3. 由父行程先取亂數隨機選擇一個位置進行炸射，此位址座標透過POSIX shared memory機制傳給子行程。子行程也透過透過POSIX shared memory回傳炸射結果給父行程，並印出炸射結果。可能的炸射結果如下：
 - a. 未擊中。
 - b. 擊中，但砲艇未沈沒。
 - c. 擊中，砲艇沈沒。判斷沈沒的條件是砲艇所有的位置均被擊中。
4. 接著輪流由雙方隨機決定位置來炸射對方砲艇，直到有一方炸沈對方的砲艇為止。當有一方砲艇被炸沈時，此時子行程結束，然後由父行程印出對戰結果。
5. 在計算過程中，父行程與子行程都要印出炸射的過程，以及自己的pid。結束的時候也要印出勝利的一方用了幾顆炸彈。

例如若父行程為 2370，子行程為2385，程式進行的一個可能的過程如下：

```
> prog1 5 10 0
```

```
[2370 Parent]: Random Seed 5
```

```
[2385 Child]: Random Seed 10
```

```
[2370 Parent]: The gunboat: (2,1)(2,2)
```

```
[2385 Child]: The gunboat: (3,2)(2,2)
```

```
[2370 Parent]: bombing (1,1)
```

```
[2385 Child]: missed
```

```
[2385 Child]: bombing (2,2)
```

```
[2370 Parent]: hit
```

```
[2370 Parent]: bombing (2,1)
```

```
[2385 Child]: missed
```

```
...
```

```
[2385 Child]: bombing (2,1)
```

[2370 Parent]: hit and sinking

[2370 Parent]: 2385 wins with 7 bombs

三、作業要點

1. 請注意，本作業使用的程式語言是C/C++，測試平台的作業系統：Ubuntu 21.10 64-bit。使用的編譯程式為gcc/g++ 編譯器：11.2。其他平台或程式語言不在本次作業考慮範圍之內。如在測試平台上無法編譯與執行，都不予給分。
2. 請注意，本作業一定要用的機制為fork()與POSIX shared memory。任何沒有使用這些機制來完成的程式，將依下列規定不予給分。
3. 父行程與子行程之間的協定以及砲艇炸射的演算法，自行規劃設計，但是地圖資訊不可以放在共享記憶體內，雙方都不能看見對方的船艦地圖。砲艇炸射過程也不可以出現作弊現象。如果有作弊現象，都不予給分。
4. 本作業的評分方式如下：
 - a. 基本功能：依照下面項目的完成程度來給分，如果只能完成部份，將部份給分。
 - i. 父行程可以從命令列正確讀取參數。本項滿分10分。
 - ii. 父行程用fork()產生一個子行程，且模擬過程中只有一個子行程。每個行程執行過程中，亂數只用給定的亂數種子設定一次。本項滿分20分。
 - iii. shared memory空間只有一塊，而且父行程與子行程只用同一塊 shared memory空間來傳遞正確的炸射結果給對方。本項滿分20分。
 - iv. 父行程與子行程有一個使用POSIX shared memory的同步機制，可以讓父行程與子行程的運作正確同步，本項滿分40分。注意，此同步機制的限制如下：
 1. 不可用到第6章的mutex機制或是任何其他IPC通訊機制，只可以使用shared memory。如果違反此限制，本項0分。
 2. 如果使用sleep()或任何暫停機制例如空迴圈的作法來避開同步問題，本項最多可得10分。
 - v. 父行程與子行程可以正確使用系統函式自己取得自己的pid。本項滿分10分。
 - b. 進階功能：完成以上基本功能且能得分達90分以上者，才可實作以下項目來得分。如果基本功能未達90分，進階功能的實作不予考慮。本項滿分30分。
 - i. 在進階功能中，子行程再用fork()產生一個新的行程，模擬三方混戰的情況。第三個行程的數種子為 $\lceil (p1+p2)/2 \rceil$ 。現在有三個行程互相輪流炸射。輪流順序是由父行程開始，接著是子行程，最後是第二個子行程。每次炸射時，分別在另兩個行程中同樣的位置進行轟炸。炸射過程中的輸出如基本的要求。炸射過程中，如果某行程的砲艇被擊沉，該行程就不能炸射，在輪流過程中必須跳過。
 - ii. 進階功能中，所有的IPC以及同步機制都必須用POSIX shared memory完成，如果使用sleep()方式或任何暫停機制，就不計分。shared memory個數則不受限制。

如果新的行程為2490，程式進行的一個可能的過程如下：

```
> prog1 5 10 1
```

```
[2370 Parent]: Random Seed 5
```

```
[2370 Parent]: The gunboat: (2,1)(2,2)
```

```
[2385 Child]: Random Seed 10
```

```
[2490 Child]: Random Seed 8
```

```
[2385 Child]: The gunboat: (3,2)(2,2)
```

```
[2490 Child]: The gunboat: (3,3)(2,3)
```

```
[2370 Parent]: bombing (1,1)
```

```
[2385 Child]: missed
```

[2490 Child]: missed
[2385 Child]: bombing (2,2)
[2370 Parent]: hit
[2490 Child]: missed
...

5. 本作業需繳交檔案：

a. 說明報告：檔案為docx或pdf格式。

- i. 報告中必須說明程式的設計理念、程式如何編譯，以及如何操作。
- ii. 報告中同時必須詳細說明你完成哪些部份。如有用到特殊程式庫，請務必說明。
- iii. 請務必讓助教明白如何編譯及測試你的程式。助教如果無法編譯或測試，會寄信（**最多兩次**）通知你來說明，但每說明一次，**助教會少給你10分**。

b. 完整原始程式碼檔案（.c 或 .cpp）。不可含執行檔。助教會重新編譯你們的程式。請注意：也不可用 .txt檔或是 .docx檔等非正常方式繳交程式碼，如有類似情形，**助教會扣10分**。

c. **不可以含有病毒，如果含有病毒等惡意程式，本作業0分。**

6. 所有相關檔案，例如報告檔、程式檔、參考資料等，請壓縮成一個壓縮檔（不可超過2MB）後上傳至portal。請注意，不可抄襲。助教不會區分何者為原始版本，被判定抄襲或雷同者，一律0分。

四、繳交方式：

1. 最終繳交時間：

a. 程式作業檔在 2022.04.01 以前，上傳至個人portal。如有多個檔案，必須將所有檔案壓縮成一個zip（rar 亦可）檔案，然後上傳。

b. 上傳檔名格式：「學號_作業號碼.rar」或「學號_作業號碼.zip」。例如：912233_01.rar 或 912233_01.zip。

2. 如有違規事項者，依照課程規定處理。

3. 如需請假，請上portal請假，並持相關證明文件，在請假結束後的第一次上課時完成請假手續，並在一週內完成補交。補交作業將以8折計算。

4. 老師不接受「門縫」方式繳交，助教也不接受任何作業。

五、如有未盡事宜，將在個人portal板面公告通知。

六、If you need **any assistance in English**, please contact Prof. Yang.

七、參考資料

1. 參考課本圖3.16與3.17。

2. 參考上課講解之範例程式及投影片的相關參考資料網頁